Fake News Detection from Data Streams

1st Paweł Ksieniewicz
Department of Systems and Computer Networks
Wroclaw Univ.of Science and Technology
Wrocław, Poland
e-mail: pawel.ksieniewicz@pwr.edu.pl
ORCID: 0000-0001-9578-8395

2nd Paweł Zyblewski
Department of Systems and Computer Networks
Wroclaw Univ.of Science and Technology
Wrocław, Poland
e-mail: pawel.zyblewski@pwr.edu.pl
ORCID: 0000-0002-4224-6709

3rd Michał Choraś
Institute of Telecommunications and Computer Science
UTP Univ.of Science and Technology
Bydgoszcz, Poland
e-mail: chorasm@utp.edu.pl

4th Rafał Kozik
Institute of Telecommunications and Computer Science
UTP Univ.of Science and Technology
Bydgoszcz, Poland
e-mail: kozikr@utp.edu.pl
ORCID: 0000-0001-7122-3306

5th Agata Giełczyk
Institute of Telecommunications and Computer Science
UTP Univ.of Science and Technology
Bydgoszcz, Poland
e-mail: agata.gielczyk@utp.edu.pl
ORCID: 0000-0002-5630-7461

6th Michał Woźniak
Department of Systems and Computer Networks
Wroclaw Univ.of Science and Technology
Wrocław, Poland
e-mail: michal.wozniak@pwr.edu.pl
ORCID: 0000-0003-0146-4205

***Abstract*—Using fake news as a political or economic tool is not new, but the scale of their use is currently alarming, especially on social media. The authors of misinformation try to influence the users' decisions, both in the economic and political sphere. The facts of using disinformation during elections are well known. Currently, two fake news detection approaches dominate. The first approach, so-called fact or news checker, is based on the knowledge and work of volunteers, the second approach employs artificial intelligence algorithms for news analysis and manipulation detection. In this work, we will focus on using machine learning methods to detect fake news. However, unlike most approaches, we will treat incoming messages as stream data, taking into account the possibility of concept drift occurring, i.e., appearing changes in the probabilistic characteristics of the classification model during the exploitation of the classifier. The developed methods have been evaluated based on computer experiments on benchmark data, and the obtained results prove their usefulness for the problem under consideration. The proposed solutions are part of the distributed platform developed by the H2020 SocialTruth project consortium.**

***Index Terms*—stream analysis, fake news, distributed architecture**

I. Introduction

The idea of using fake news to achieve political or economic goals is not new and was used, among others, in ancient Rome, where Octavian spread false information about Mark Antony, who was allegedly a drunkard, a womanizer and a toy in the hands of Queen Cleoparty[1]. We may also read how the disinformation could be used in the Holy Bible, e.g., in the resurrection description[2].

Unfortunately, for several years the problem of using fake news has become more and more nagging and recently concerned the presidential campaigns in the USA and France. Fake news could also be used as a political weapon, which may promote desirable ideas or behaviours on the ground. The Russian Federation widely uses it as part of the information war, which has been followed by an EU response, resulting in the creation of the EUvsDisinfo website[3] to monitor and counteract disinformation campaigns.

Another example is the systematic activity of the Lithuanian government, which allows temporary shutting down servers for 48 hours without a court order if they are used to spread fake news[4]. These activities are supported by civic movements, the so-called net of elves, which manually check the news published on the Internet and report the detected violations to the authorities.

Currently, more and more websites are being created that are trying to help assess the accuracy of the information, the so-called fact-checkers. Unfortunately, it is not possible to check them manually with such a massive amount of new messages. Hence, more and more hope is placed in automatic fake news

[1] Kaminska, Izabella, A lesson in fake news from the info-wars of ancient Rome, Financial Times, January 17, 2017, https://www.ft.com/content/aaf2bb08-dca2-11e6-86ac-f253db7791c6?mhq5j=e3

[2] ”So the soldiers took the money and did as they were instructed. And this story has been widely circulated among the Jews to this very day.” (Matthew 28:15)

[3] https://euvsdisinfo.eu/

[4] Michael Peel, Fake news: How Lithuania's 'elves' take on Russian trolls, Financial Times, Feb. 4, 2019, https://www.ft.com/content/b3701b12-2544-11e9-b329-c7e6ceb5ffdf

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

detection systems using machine learning methods. The main reason for the rapid increase in the use of disinformation is the ability to use not only traditional mainstream media but also social media, like Twitter or Facebook. It is worth noting that its popularity can rapidly grow according to the rule that *false news spreads faster and more comprehensive*. Its extensive spread has a severe negative impact on media users and society.

The main goal of publishing such information with malicious content is to attract readers, which could increase publisher rank and popularity, which consequently increases revenues form adds. It is worth mentioning that there is no single definition of what fake news is and what it is not. Shu et al. proposed the following taxonomy [1]:

- satire news with proper context and hoaxes,
- rumours,
- conspiracy theories,
- misinformation that is created unintentionally.

Interesting are studies analyzing which types of attributes prove to be useful in the context of fake news classification [1]. Among the standard solutions present in the research, we may distinguish the analysis of creators and readers of texts, document content, stylometric analysis and verification of positioning the document in social networks [2]. Image analysis that focuses on false video information is also a promising approach [3]. Equally interesting seems to be the work of Conroy at al. [4] proposing a distinction between approaches to linguistic (semantic, rhetorical, discourse and simple probabilistic recognition models) and social (analysis of the author's behaviour within the social network or analysis of the built context by all of his posts). Castillo et al. [5] bases the construction of recognition models on users' behaviour in the context of posting and forwarding content depending not only on their content but also references to other documents. Ferrara et al. [6] analyze various data representations, and Afroz et al. [7] different variations of stylometric metrics.

Sharma et al. [8] discussed several topics related to the problem in question and pointed to the possibility of using the SCAN (Scientific Content Analysis) method to solve it. Jin et al. [9] proposed the compelling approach for the task of automatic news verification, departing from text analysis in favour of image data. Zhang et al. [2] pointed to the dynamic nature of social media messages and proposed analyzing them in the context of streaming data. Horne and Adali [10] employed SVM to distinguish between fake, authentic and satire messages. This kind of classifier has also been used by Cheng et al. for users classification, using semantic analysis and behavioural feature descriptors to detect potentially fake online posts [11].

Interesting comparative studies by Gravanis et al. [12] evaluated several linguistic features based classification approaches. They presented the results showing that known classifier models, especially ensembles, may be successfully used as fake news detectors. Bondielli et al. [13] pointed out that while anomaly detection and clustering methods could be used for fake news detection, this problem is usually reduced to the classification task. Atodiresi et al. [14] considered an NLP tools-based approach to tweet analysis. The authors defined this problem as a regression task and thus were able to assign a credibility value to each message.

Unfortunately, in most works, the authors consider the problem of fake news detection is a classic problem of data analysis, without taking into account their streaming nature. What is more, it should be taken into account that the profile of messages classified as fake news may change over time, i.e., we are dealing with a phenomenon known as concept drift. This is since, as with other information security problems, such as the detection of unwanted mail, the authors of fake news are aware that publishing them is becoming more difficult because automatic detection systems will detect them. Thus, it can be expected that their profile will change over time to deceive these systems, and therefore requires authors of these types of systems to equip them with mechanisms for adapting to changes in probabilistic characteristics of the fake news detection task. This work will attempt to develop fake news detection algorithms based on a data stream where we will not assume its stationary nature [15]. To the best of the authors' knowledge, there is no work treating fake news detection as a problem of streaming data classification. Although some authors note that social media data are of such nature, only Wang and Terano [16] use techniques adequate to analyze data streams. However, their approach is limited to relatively short streams and does not potentially take into account the non-stationary nature of the data.

The main contributions of this paper are as follows:

- Formulating the problem of fake news detection as a data stream classification task.
- Proposing a novel pattern classification methods based on feature extraction techniques, which address the detection of fake news in streaming data from social media.
- An extensive experimental analysis backed-up by the statistical tests.

II. SocialTruth Platform Architecture

The proposed machine learning solutions constitute text verification services, one of the critical elements of the *SocialTruth* platform. From a broader perspective, it is crucial to explain the environment where the proposed solutions will operate and how they will bring benefits for the end-users, which are all kind of actors that need to cope with fake news challenges.

Therefore, in this section, we give a general overview of the *SocialTruth* project platform, which has been depicted in Figure 1. The technology stack has been decomposed into the following logical elements that have been detailed in the next subsections:

- physical elements (nodes) and their orchestration,
- verification services,
- messaging and event processing.

The data is ingested into the platform either by the user (journalist, author, reader) over HTTP(S) protocol or using dedicated crawlers (data connectors) that send data over the binary protocol to the *Apache Kafka* framework. The *Apache Kafka* is a distributed streaming platform implementing the publish-subscribe model. Once the ingested data is published to one of the *Kafka* topics, it can be simultaneously consumed by various verification services and stream processing applications. Once the services When the services finalize their computations, they make the results available on another *Kafka* topic, which can be consumed by other services again. Such kind of processing pipeline is called choreography pattern. When the services finalize their computations, they make the results available on another *Kafka* topic, which can be consumed by other services again. Such kind of processing pipeline is called choreography pattern. It is a contradiction to the orchestration pattern, which introduces a central entity (orchestrator) controlling the execution of each stage in the pipeline. These two approaches have their advantages. We use a mix of both when implementing web service handling the HTTP requests.

A. Physical nodes comprising the system

The first and the most bottom layer in the technology stack constitutes the orchestration framework. It is laid down on top of an infrastructure composed of virtual and hardware machines. This layer is intended to implement automated resource management, and thus it facilitates the entire platform with such capabilities as flexibility, scalability, and fault tolerance. It is the responsibility of the orchestration layer to effectively deploy the services on the available computational nodes (both physical and virtual). It is achieved thanks to containers that are sandboxes that contain the implemented service together with all the software dependencies (libraries and execution environment). In such a form, the services can be easily migrated between the computational nodes and deployed. In the proposed solution, we have used *Docker Swarm* system.

B. Verification services

All the fake news detection mechanism (presented in this paper) are the instances of text verification services. As such, they are the critical building blocks of the system and are deployed as a micro-services. Micro-service is an independently deployable component, which (in the proposed architecture) is packed as a *Docker* container. Each micro-service is focused on providing single functionality. Moreover, each functional service provides an API that allows other clients to interact with the service in synchronous (e.g. REST) and asynchronous ways (e.g., events). Each service can also have client API for interacting with other components/services (e.g., databases). Moreover, the service can subscribe (listen) to a notification sent from other components in the system. In the proposed architecture, we heavily use asynchronous event-based communication in favour of synchronous calls. This allows us to avoid tight coupling between the verification services and other components in the platform. In that regard, each verification service subscribes to a dedicated topic and produces results on another one. In the diagram of the architecture, we deliberately depicted image and meta-verification services. These constitute essential elements of the platform as news commonly is accompanied by images to support the content. However, this matter is out of the scope of this paper.

C. Messaging and event processing

As we mentioned before, in the described system, we have adapted *Apache Kafka*. It is a distributed streaming platform, which enables both real-time event processing and event-driven communication between various components. From the architectural point of view, *Apache Kafka* constitutes a flexible and efficient way to integrate all the components, both existing tools as well as the new ones developed during the project or by the community.

Moreover, on top of the *Apache Kafka* system, we deploy the *Complex Event Processing* (CEP) engine. We use it to pre-process the data before storing and presenting it to the end-user or the verification service. For example, we use this technology to join data streams produced by verification services belonging to the same type. In that regard, we can present the user analysis results obtained from different classifiers.

III. Methods

As we noticed in the introduction to the following work, the overwhelming majority of machine learning research in the field of fake news detection relies on the extraction of linguistic features. The main subject covered in the study presented in this work is, however, the analysis of approaches to feature extraction for the needs of the data stream classification task for the problem above. At the entry point, each of the patterns constructing the stream, representing individual articles, contains a thousand attributes determined by a simple solution of *Count Vectorizer* – applied to isolate the impact of diverse linguistic analysis methods on the quality of results achieved.

The dataset developed for the experiment is based on the *Getting Real about Fake news* set, containing 13,000 articles marked as fake news by *BS Detector Chrome Extension* users. In each of them, we have the title, content and timestamp. To develop the classification model, it was supplemented with the same number of articles from sources commonly considered to be verified and reliable, selected in a similar period indicated by timestamps of data from the original set. The dataset developed in this way was then ordered following the publication dates, to develop a data stream enabling to perform the reliable experiments.

A. Dimensionality reduction

A thousand attributes of a learning set constitute a problem of very high dimensionality. Many features may show mutual statistical dependence, and many of them may prove to be utterly unimportant in the context of the classification problem under consideration, showing no relationship with the actual problem labels. It turns out that it is necessary to reduce the scale of the problem or modify it using methods of

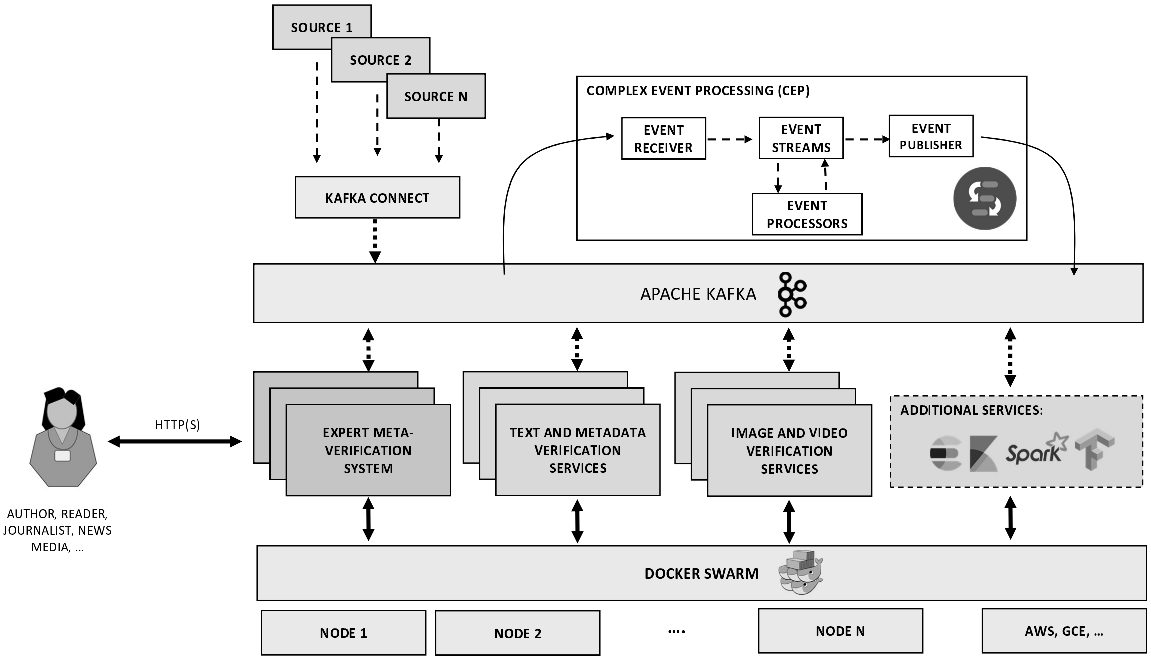


Fig. 1: SocialTruth Platform - general overview of the architecture and technology stack

feature extraction and selection. In the conducted experimental evaluation, three strategies of this type were applied.

a) PCA*:* The first considered strategy was the *Principal Components Analysis*. For the set of observations, the coordinate system is rotated in such a way as to maximize the variance of subsequent attributes. It leads to an increasing percentage of the explained variance. It allows the transformation of data to representations with a lower number of features than the input set by the combination of real attributes of the problem. It is a state-of-art solution to extract features for multidimensional data.

b) COUNT VECTORIZER*:* The second method used in experiments is based on the base approach of feature extraction – *Count Vectorizer*. To take into account the same impact of document titles and content, each of the subsets of features was normalized using the *standard normalization*, followed by the selection of the features represented by the most significant number within the data used to construct the extraction model.

c) FEATURE SELECTION*:* The last method used was a filter-based feature selection. Methods from this group use statistical techniques to assess the relationship between each feature and problem classes [17]. The scores obtained are then used to select the most significant features. In this case, as the correlation measure, the *Chi-Squared test* was used.

Each of the considered methods of reducing the feature space, due to the streaming nature of the processing, was implemented in the form of a model fitted based on the first portion of data supplied to the classification system (the first chunk). Subsequent portions of data were reduced under the model prepared in this manner, so they showed the same, processed characteristics. However they did not pose a problem of *data peeking*, due to the lack of influence on the feature extraction model.

B. Strategies to train the classifier on a data stream

Each of the three proposed dimensionality reduction strategies was analyzed in three different *state-of-art* methods for constructing classification models in data streams.

*a) Streaming Ensemble Algorithm (*SEA*):* Proposed by Street and Kim in [18], SEA constructs a classifier ensemble of a fixed size, by training a new base classifier on each observed data chunk. This approach is separate from the commonly used approaches with updated models [19]. In case of exceeding the fixed pool size, the worst performing model according to a given metric is removed. The final decision of the ensemble is produced according to the sum rule [20].

*b) Online bagging (*OB*):* Ensemble learning algorithm proposed by Oza in [21] and based on offline *Bagging*. It maintains a classifier pool in which, with the arrival of the new sample, each base estimator is trained on it K times, where K comes from the $Poisson(\lambda = 1)$ distribution.

*c) Single model (*SM*):* In addition to the classifier ensembles, the natural ability of selected classifiers to adapt to partial fitting was also tested, where a single model is constructed. However, each incoming data chunk is used to modify its properties with knowledge acquired based on a new class distribution. This approach, apart from the classifiers that effectively provide the forgetting mechanisms, is not immune to the concept drift phenomenon. Nevertheless, unlike ensemble methods, it is not strongly dependent on the size of a single chunk of data used in processing [22].

C. Base classifiers for data stream processing

Each of these processing methods also requires the selection of a base classifier, which – due to the consideration of processing using single models – must meet the requirement to be able to conduct a partial fit of the already built recognition model. Three classification algorithms meeting this condition were selected.

GNB *Gaussian Naive Bayes* – without prior probabilities,

MLP *Multi-layer Perceptron* – with one hidden layer build on 100 neurons, using ReLU activation function and stochastic gradient-based optimizer.

HT *Hoeffding Tree* – using *gini* split criterion and *Naive Bayes Adaptive* prediction mechanism.

IV. Experimental setup

The entire experimental evaluation was implemented using *Python* libraries, based on the *scikit-learn* [23] module in the implementation of two base classifiers and all feature reduction methods, on the *stream-learn* [24] module in data stream processing, calculating evaluation metrics and employed classifier ensembles of stream processing and on the *scikit-multiflow* [25] module in the implementation of *Hoeffding Tree*.

The implementation of the analyzed processing methods, supplemented with a module of datastream generation prepared following the description of static data included in Section II, together with the analytical script used to generate all tables and illustrations contained in the following section, is publicly available on the GIT repository (https://github.com/w4k2/fakestreams).

During the process of methods evaluation, the *state-of-art Test-Then-Train* methodology was used, which involves alternating testing of algorithms on an incoming portion of data, which has not yet been made available to the classifier for the needs of learning and updating its model after supplementing it with original labels. Two hundred fifty patterns were adopted as the size of a single chunk.

Each of the three feature extraction methods has been paired with each of the three stream processing strategies built on each of the three considered base classifiers, assuming 2, 10, 50, 100, 200, 500 or 1000 extracted attributes for the construction of the classification model, which resulted in 189 runs being the basis of the evaluation. Due to the balanced nature of the problem, the results are presented using the accuracy metric, being appropriate for this kind of data.

V. Experimental evaluation

The results of the conducted experiments were collected in Table I, divided by horizontal sections into individual stream processing strategies, with the numbers of extracted features and vertical sections for the used spatial reduction methods. Besides, in Figures 2, 3 and 4 respectively, the experiments' runs for each of the processing strategies were presented. The illustrations were reduced to runs only for 10, 100, and 1000 attributes to increase readability.

The results show that the FS method is the most effective when paired with SEA. At the same time, the MLP, in this case, is characterized by a gradual increase in the classification accuracy until 200 attributes, but over time reducing its performance. The behaviour of HT may be unusual, which despite initial growth, after exceeding 500 features, decreases in generalization capacity to the random classifier level. A similar observation is valid for PCA and CV. When the GNB classifier is used as the base for SEA, the method starts from a level slightly higher than the random classifier and achieves decent classification accuracy over time, but not at a level comparable to MLP.

The OB strategy seems to be far less suited for reduction by FS than SEA. The best results are achieved, again by the MLP classifier used as the base, using the PCA method. In this case, for a combination of PCA, OB and MLP with 1000 attributes generated, we achieve the highest average classification value of 81 per cent among the experiments carried out. An interesting difference between the SEA and OB approaches is that GNB rarely goes beyond the level of the random classifier for the latter and PCA reduction. On the other hand, in the case of CV and FS, GNB is characterized by a steady increase in classification ability, directly dependent on the number of problems features. HT properties do not differ from those developed by the SEA strategy.

The use of SM in the classification reveals the weakness of the PCA algorithm in tandem with GNB, further reducing the quality of such a solution. However, it can be seen that the same reduction with the MLP algorithm leads to the best results among all SM-base strategies. In this case, the CV extraction does not meet the expectations, in any of the tested instances leading to the best solution, which is evenly distributed between PCA and FS.

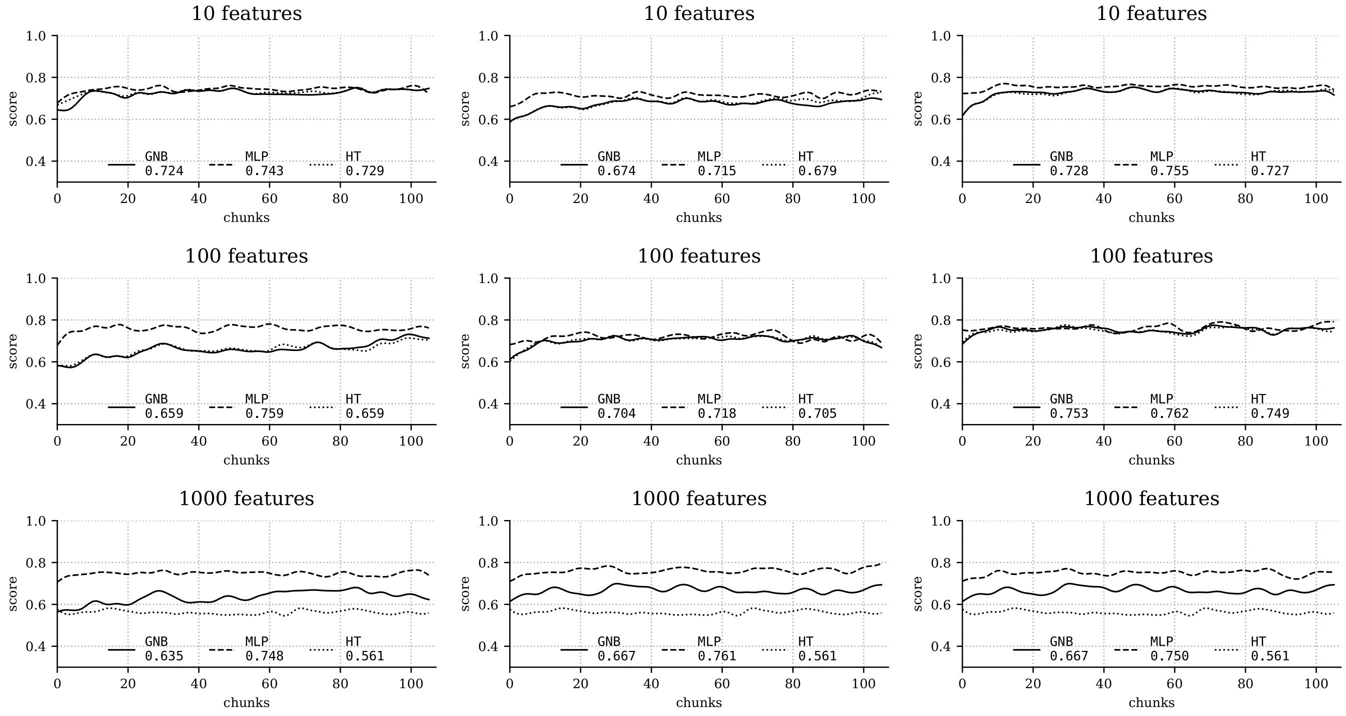
Summing up the analysis of the results obtained, it can be stated that the most effective of the considered classification algorithms is MLP. One may see a simple linear relationship between its generalization ability and the attributes number of the constructed model, almost regardless of the used extraction method and processing strategy. Obvious observation for all approaches is also quite the inverse relationship that occurs for HT, whose quality of classification degenerates with the increase in the dimensionality of the problem. GNB and PCA can be considered as the worst combination of streaming approaches, especially for OB and SM. MLP classifier with OB and PCA should be distinguished as the best combination of all analyzed strategies to deal with dimensionality reduction in fake news data stream processing.

VI. Conclusions

It is one of the first work which treats the problem of fake news detection as the stream data classification task and also takes into consideration that the characteristics described the incoming messages could change over time. Studies of this type have not been represented widely in the literature so far,

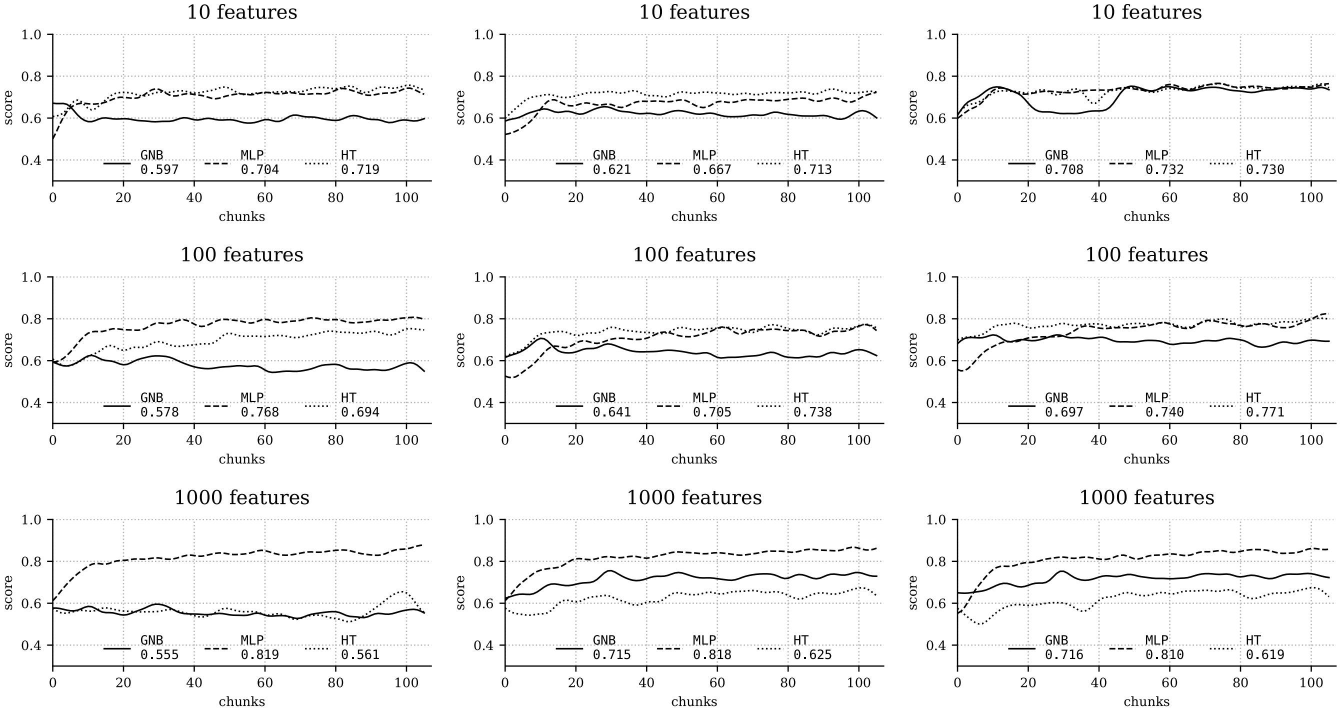
TABLE I: Results of experimental evaluation.

FEATURES	PCA			COUNT VECTORIZER			FEATURE SELECTION		
	GNB	MLP	HT	GNB	MLP	HT	GNB	MLP	HT
	SEA								
2	0.602	0.666	0.570	0.553	0.628	0.560	0.663	**0.679**	0.673
10	0.724	0.743	0.729	0.674	0.715	0.679	0.728	**0.755**	0.727
50	0.693	0.770	0.701	0.714	0.728	0.717	0.744	**0.778**	0.744
100	0.659	0.759	0.659	0.704	0.718	0.705	0.753	**0.762**	0.749
200	0.656	0.692	0.671	0.716	0.733	0.693	0.758	**0.764**	0.713
500	0.664	**0.747**	0.561	0.732	0.733	0.561	0.745	0.746	0.561
1000	0.635	0.748	0.561	0.667	**0.761**	0.561	0.667	0.750	0.561
	ONLINE BAGGING								
2	0.516	0.641	0.664	0.517	0.604	0.612	0.636	0.677	**0.688**
10	0.597	0.704	0.719	0.621	0.667	0.713	0.708	**0.732**	0.730
50	0.613	**0.768**	0.711	0.657	0.695	0.745	0.706	0.728	0.767
100	0.578	0.768	0.694	0.641	0.705	0.738	0.697	0.740	**0.771**
200	0.569	0.754	0.658	0.650	0.758	0.708	0.691	**0.764**	0.738
500	0.559	**0.804**	0.589	0.673	0.785	0.633	0.722	0.792	0.616
1000	0.555	**0.819**	0.561	0.715	0.818	0.625	0.716	0.810	0.619
	SINGLE MODEL								
2	0.512	0.633	0.654	0.518	0.595	0.611	0.634	0.673	**0.690**
10	0.595	0.700	0.702	0.622	0.662	0.707	0.710	**0.727**	0.694
50	0.618	0.768	0.672	0.644	0.697	0.730	0.707	0.727	**0.747**
100	0.579	**0.766**	0.663	0.645	0.705	0.722	0.696	0.742	0.745
200	0.562	0.752	0.667	0.651	0.753	0.714	0.686	**0.762**	0.724
500	0.556	**0.796**	0.618	0.672	0.777	0.661	0.719	0.788	0.630
1000	0.554	**0.808**	0.615	0.722	0.806	0.626	0.722	0.805	0.626



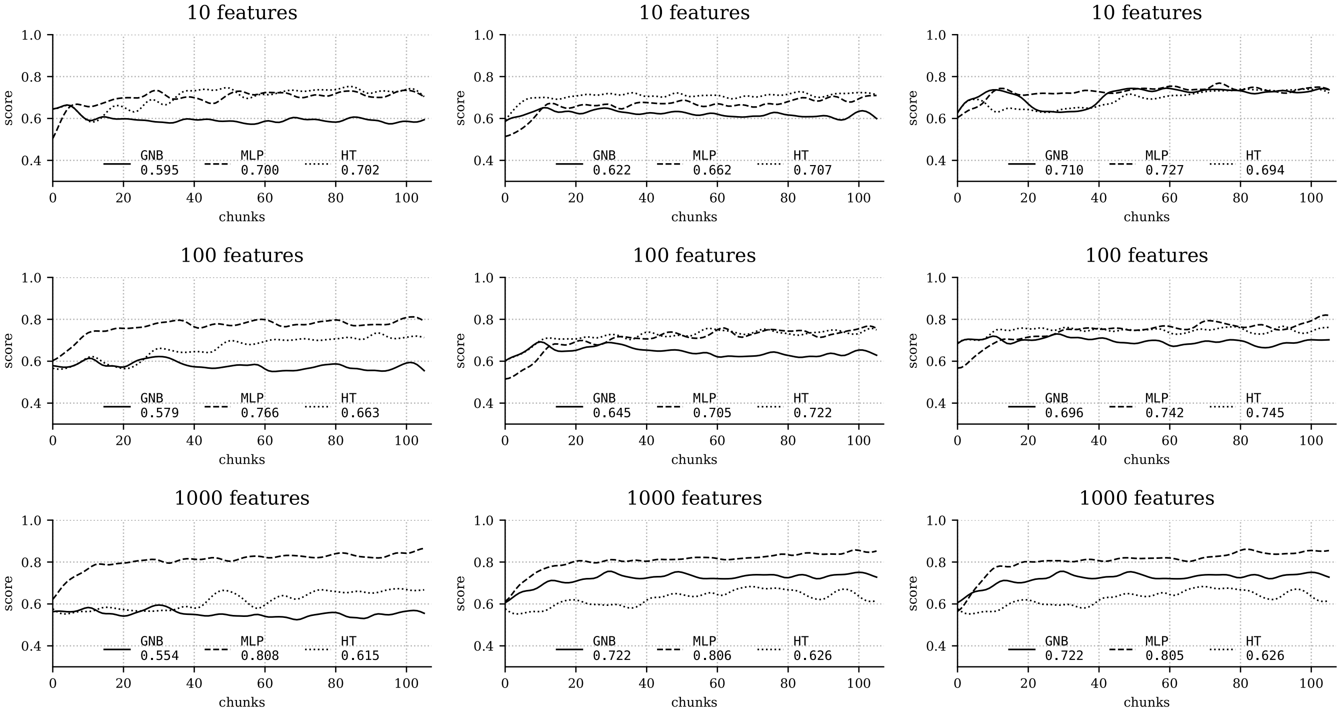
(a) PCA (b) COUNT VECTORIZER (c) FEATURE SELECTION

Fig. 2: Example runs of data stream processing for SEA processing approach.



(a) PCA (a) COUNT VECTORIZER (c) FEATURE SELECTION

Fig. 3: Example runs of data stream processing for ONLINE BAGGING processing approach.



(a) PCA (b) COUNT VECTORIZER (c) FEATURE SELECTION

Fig. 4: Example runs of data stream processing for SINGLE MODEL processing approach.

so it is a preliminary analysis of the effectiveness of typical methods of feature reduction and the construction of stream models for an entirely new problem.

Extensive experimental research on several algorithms from each of the considered aspects of processing confirmed the possibility of constructing systems of this type in an application for data with a stationary nature of the concept, suggesting extraction using the *Principal Components Analysis* algorithm in the construction of *Online Bagging* ensemble encapsulating the *Multi-layer Perceptron* base classifier.

The research presented in work will be continued by considering subsequent analyzes also data dynamics both in prior and posterior probabilities, taking into account different variants.

Acknowledgement

This work is funded under SocialTruth project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825477.

References

[1] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *SIGKDD Explor. Newsl.*, vol. 19, no. 1, pp. 22–36, Sep. 2017. [Online]. Available: http://doi.acm.org/10.1145/3137597.3137600

[2] X. Zhang and A. A. Ghorbani, "An overview of online fake news: Characterization, detection, and discussion," *Information Processing & Management*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457318306794

[3] M. Choraś, A. Giełczyk, K. Demestichas, D. Puchalski, and R. Kozik, "Pattern recognition solutions for fake news detection," in *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, 2018, pp. 130–139.

[4] N. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, vol. 52, pp. 1–4, 01 2015.

[5] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *Proceedings of the 20th International Conference on World Wide Web*, ser. WWW '11. New York, NY, USA: ACM, 2011, pp. 675–684. [Online]. Available: http://doi.acm.org/10.1145/1963405.1963500

[6] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, pp. 96–104, Jun. 2016. [Online]. Available: http://doi.acm.org/10.1145/2818717

[7] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 461–475. [Online]. Available: https://doi.org/10.1109/SP.2012.34

[8] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, "Combating fake news: A survey on identification and mitigation techniques," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 3, pp. 21:1–21:42, Apr. 2019. [Online]. Available: http://doi.acm.org/10.1145/3305260

[9] Z. Jin, J. Cao, Y. Zhang, J. Zhou, and Q. Tian, "Novel visual and statistical image features for microblogs news verification," *Trans. Multi.*, vol. 19, no. 3, pp. 598–608, Mar. 2017. [Online]. Available: https://doi.org/10.1109/TMM.2016.2617078

[10] B. D. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news," *CoRR*, vol. abs/1703.09398, 2017. [Online]. Available: http://arxiv.org/abs/1703.09398

[11] C. Chen, K. Wu, S. Venkatesh, and X. Zhang, "Battling the internet water army: Detection of hidden paid posters," *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pp. 116–120, 2011.

[12] G. Gravanis, A. Vakali, K. Diamantaras, and P. Karadais, "Behind the cues: A benchmarking study for fake news detection," *Expert Systems with Applications*, vol. 128, pp. 201 – 213, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417419301988

[13] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Information Sciences*, vol. 497, pp. 38 – 55, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025519304372

[14] C.-S. Atodiresei, A. Tănăselea, and A. Iftene, "Identifying fake news and fake users on twitter," *Procedia Computer Science*, vol. 126, pp. 451 – 461, 2018, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050918312559

[15] P. Ksieniewicz, M. Woźniak, B. Cyganek, A. Kasprzak, and K. Walkowiak, "Data stream classification using active learned neural networks," *Neurocomputing*, vol. 353, pp. 74–82, Aug. 2019. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.05.130

[16] S. Wang, L. L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, 2015.

[17] P. Ksieniewicz and M. Woźniak, "Imbalanced data classification based on feature selection techniques," in *Intelligent Data Engineering and Automated Learning – IDEAL 2018*. Springer International Publishing, 2018, pp. 296–303. [Online]. Available: https://doi.org/10.1007/978-3-030-03496-2_33

[18] N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," *Proceedings of the 7Th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 377–382, 01 2001.

[19] A. Cano and B. Krawczyk, "Kappa updated ensemble for drifting data stream mining," *Machine Learning*, vol. 109, no. 1, pp. 175–218, Oct. 2019. [Online]. Available: https://doi.org/10.1007/s10994-019-05840-z

[20] R. P. W. Duin, "The combining classifier: to train or not to train?" in *Object recognition supported by user interaction for service robots*, vol. 2, Aug 2002, pp. 765–770 vol.2.

[21] N. C. Oza, "Online bagging and boosting," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, Oct 2005, pp. 2340–2345 Vol. 3.

[22] P. Zyblewski, P. Ksieniewicz, and M. Woźniak, "Classifier selection for highly imbalanced data streams with minority driven ensemble," in *Artificial Intelligence and Soft Computing*. Springer International Publishing, 2019, pp. 626–635. [Online]. Available: https://doi.org/10.1007/978-3-030-20912-4_57

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] P. Ksieniewicz and P. Zyblewski, "stream-learn–open-source python library for difficult data stream batch analysis," *arXiv preprint arXiv:2001.11077*, 2020.

[25] J. Montiel, J. Read, A. Bifet, and T. Abdessalem, "Scikit-multiflow: A multi-output streaming framework," *Journal of Machine Learning Research*, vol. 19, no. 72, pp. 1–5, 2018. [Online]. Available: http://jmlr.org/papers/v19/18-251.html